SEQUENTIAL LOGIC INSTRUMENTATION

by

WILLIAM T. HULL

B. S. (E. E.)
Kansas State University, 1963

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
MANHATTAN, KANSAS

1964

Approved by:

Charles H. Manich
Major Professor

TABLE OF CONTENTS

INTRODUCTION

A sequential logic machine may be defined as a device which performs a prescribed discrete operation on one or more variables in a certain sequence that is dependent upon time, t. This may be expressed by

$$F_n = S_{n-1}(t_{n-1}, F_{n-m}, x_j, a_k) \qquad 1)$$

where

F_n = present discrete output

S_{n-1} = previous input

F_{n-m} = any previous outputs

x_j = external inputs

a_k = internal machine constants

t_{n-1} = time

It is to be noted that F_n as well as S_{n-1} can be a combination of several outputs and inputs respectively. The solution of the above expression in terms of physical logic elements is the task to which this paper is directed.

An example of the above-mentioned machine is a general purpose digital computer. The output of the computer is dependent upon time, previous outputs, external inputs, and machine constants. The outstanding fact is that the output does not reach its prescribed value the instant the input (S_{n-1}) is applied, but rather, goes through a transition period of finite duration. The digital computer, through its "stored program", may select alternate transitions (S_{n-1}'s) which lead to varied outputs. This ability to select its transitions distinguishes the digital computer from its counterpart the desk calculator (man must supply the decision of selecting the desired transition). The transition is defined by Caldwell (1) as an unstable state.

In realizing an interconnection of logic elements to simulate equation 1 (which is the fundamental design task) it is apparent that a general solution is virtually impossible. This paper, therefore, deals with a specific finite system. It should not be implied that the design procedure to be outlined will work for any and all sequential machines, but it should be noted that the reasoning and ideas behind the procedure will enable the designer to generate a great variety of systems.

There have been proposed, by many people, numerous procedures for sequential design. Caldwell (1), Marcus (2), and Phister (3) have all approached the design problem from slightly different perspectives; each has reached the same goal, a system that works. The design procedure to be formulated will draw on the outstanding features from the works of the three previously mentioned writers.

It is important to note throughout the text that two-valued Boolean positive logic is used exclusively.[1] Further explanations and symbols used can be found in Appendix A.

[1] Reference 3, p.31.

MOTIVATION

In the summer of 1963, an instrumentation requirement was specified by the Department of Psychology of Kansas State University which was eventually satisfied through the design and implementation of a pulsed input sequential logic circuit, and operational amplifiers.

Experiments were being performed by the Department of Psychology on human subjects to determine the effect of variations of several parameters on classical conditioning.[1] The method of data collection and analysis pertaining to the experiment was hand scoring of a strip chart recording. The information recorded on the strip chart was a continuously varying signal which was proportional to the eye-lid position of the subject. (See Fig. 1) Initially the pertinent data taken from this recording was the area beneath the variable signal and above the baseline, and the time, measured from some arbitrary reference, at which the random signal crossed the baseline. The integration was being done by hand counting squares and the time was taken as linear distance. Time, cost, and susceptibility to error using these techniques were prime motives in designing and constructing an electronic system to perform the indicated tasks.[2]

[1] Kansas State University Department of Psychology proposal to U. S. Department of Health and Welfare - MH07643-01, dated October 30, 1962.

[2] Ibid.

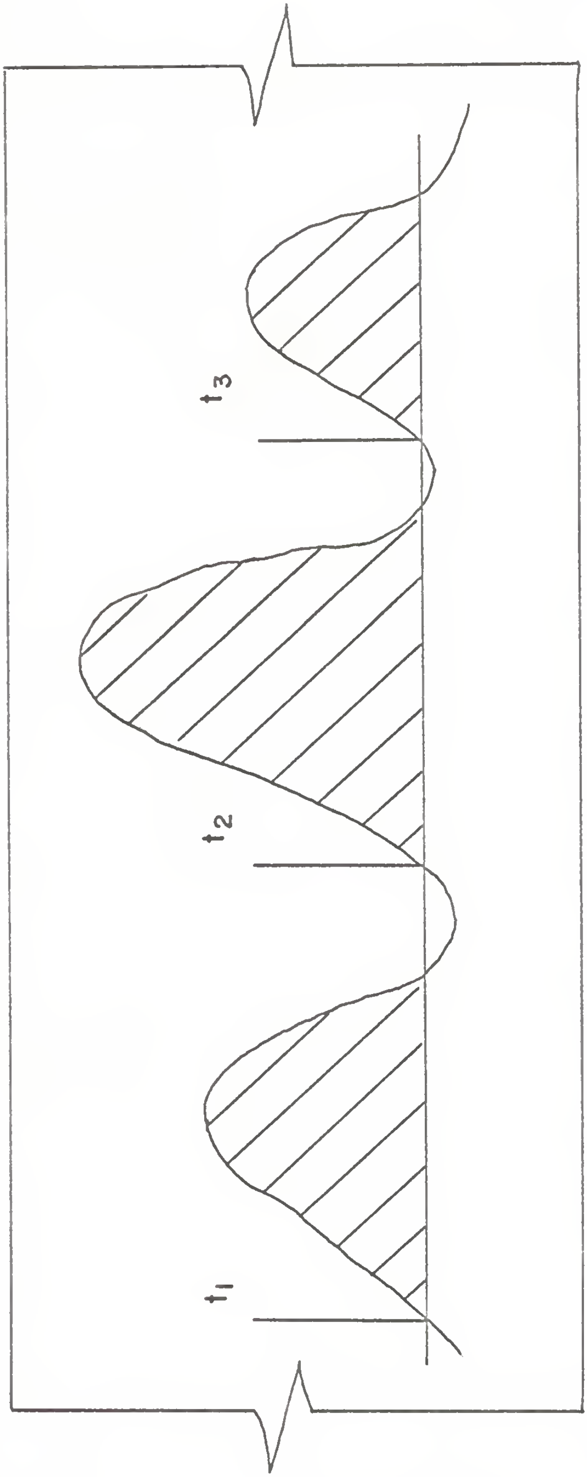


FIG. I. STRIP CHART RECORDING

DESCRIPTION OF ORIGINAL SYSTEM

The original system consisted of two major elements, viz., a programming unit and a recording-scoring unit as shown in Fig. 2. The function of the programming unit was to produce and vary the conditioning variables. It consisted of pulse and waveform generators. Of prime concern was the recording-scoring unit which was made up of a potentiometer (which attaches to the eye-lid) in the input circuit of the commercially available "eye-blink" amplifier, and a recorder on which the pertinent data was recorded. The output of the "eye-blink" amplifier (E_e in Fig. 3) will be the input signal to the system to be constructed.

SYSTEM DESIGN

Modification of Input Signal

As can be seen (in Fig. 3) the voltage output of the "eye-blink" amplifier is above ground (approximately 120 volts). It was desired to lower the baseline to ground level. This was accomplished by the use of an operational amplifier network shown in Fig. 4. The following is an explanation of the circuit action.

When the eye-lid is open (which is the normal position), $I_1 = I_2$ and the output $E_a = 0.0$. When the eye-lid closes, I_1 increases as I_2 decreases (a proportional amount) causing

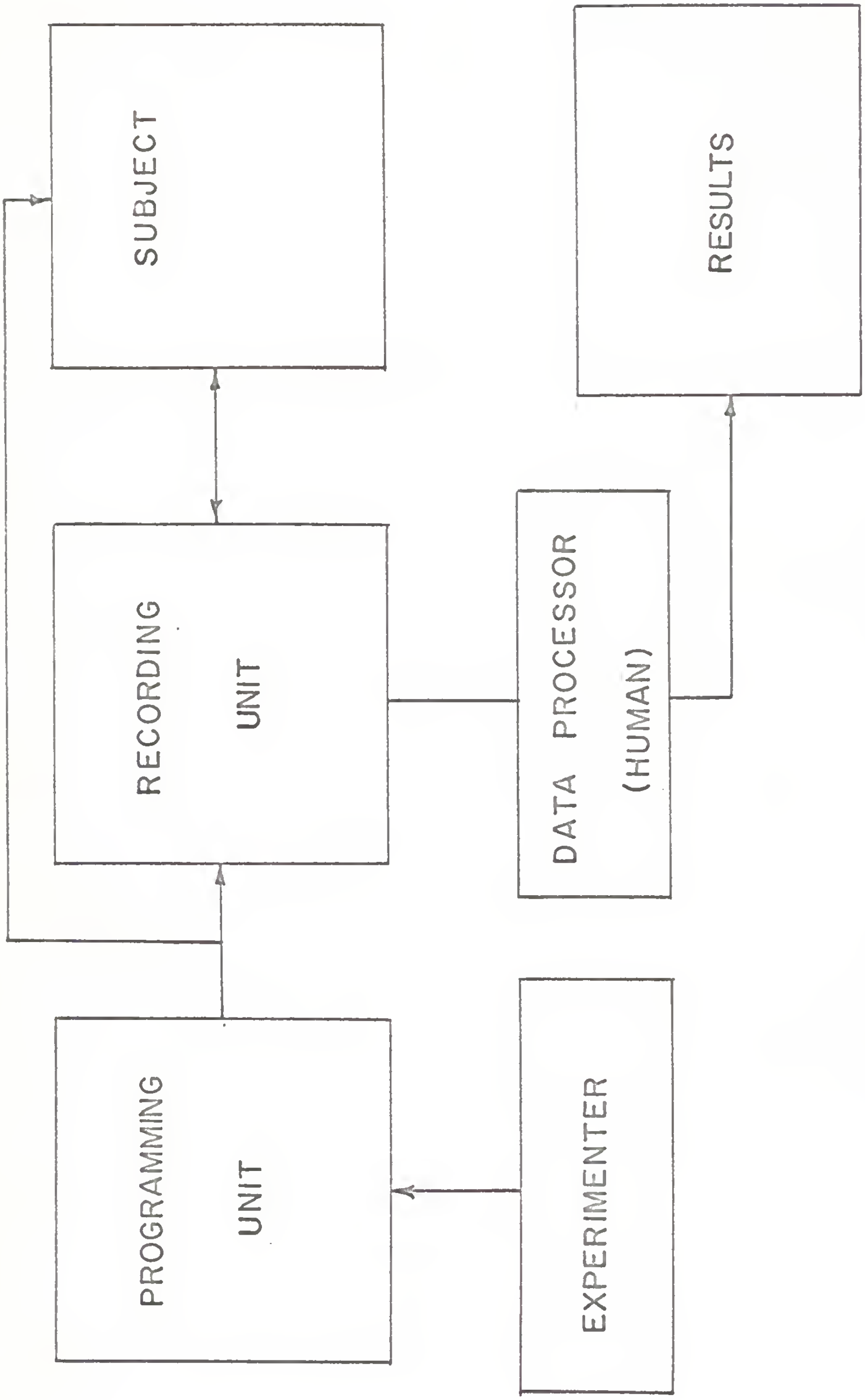


FIG. 2. EXISTING SYSTEM

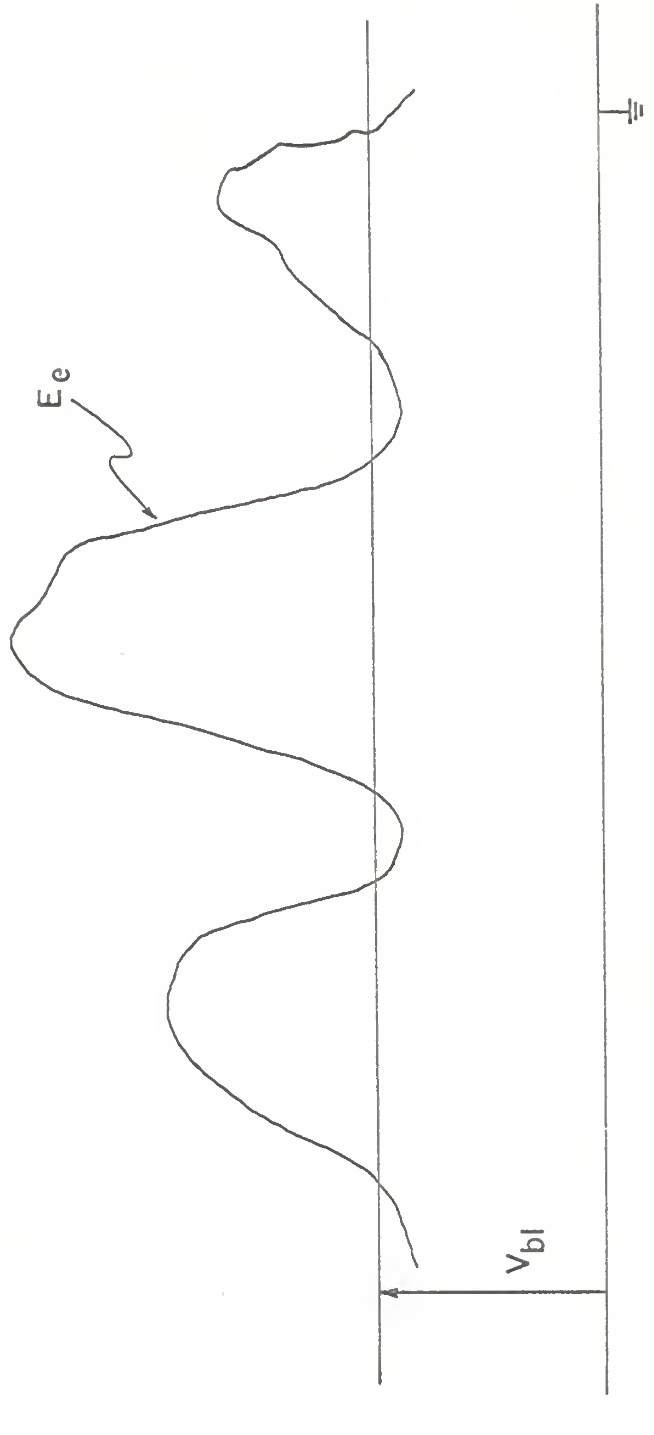


FIG. 3. INPUT SIGNAL

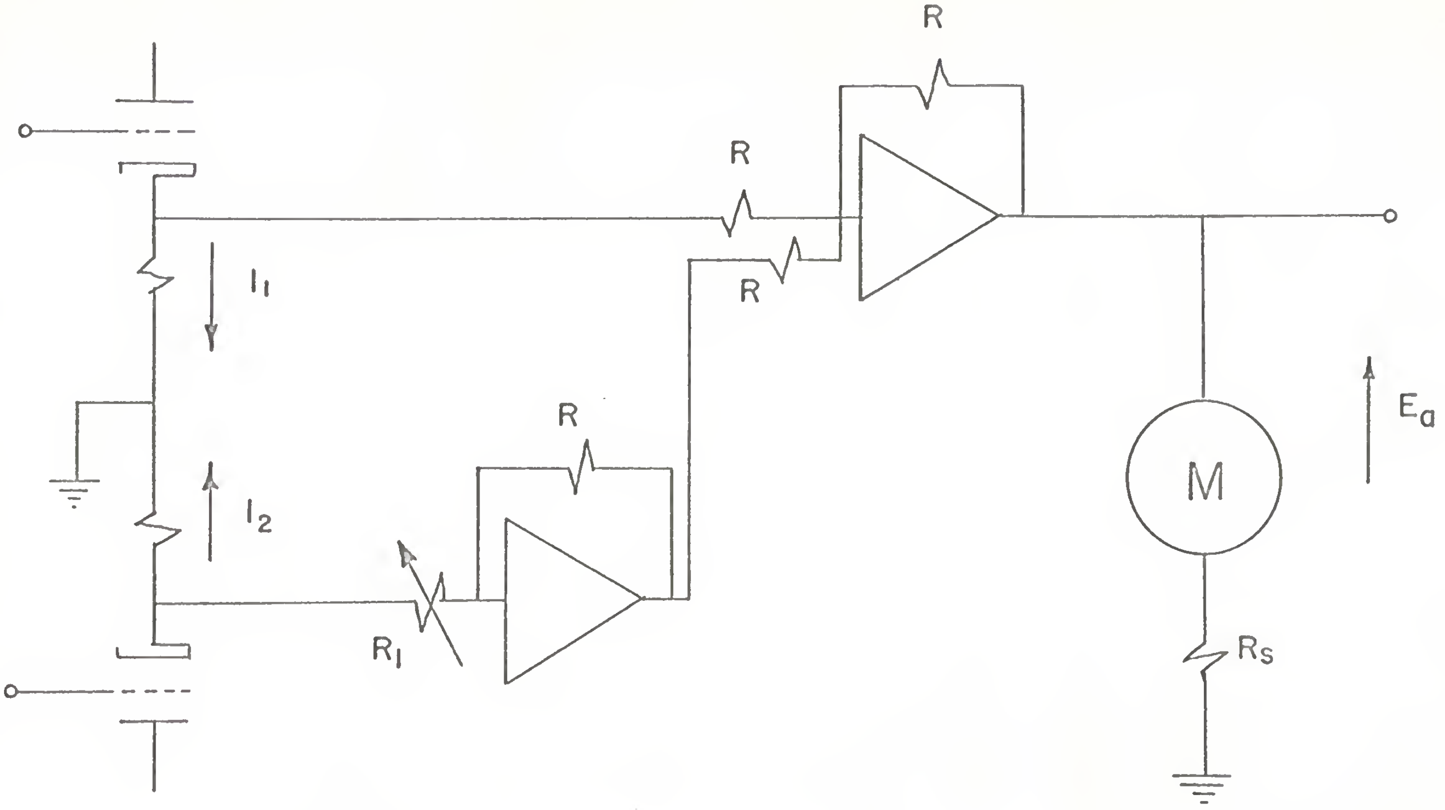


FIG. 4. OPERATIONAL AMPLIFIER NETWORK

$-E_e-KE_e-V_{bL}+V_{bL}$ to appear at the output, where $K = R/R_1 \doteq 1$. This voltage (E_a) was used as the signal to be integrated; it was also used to generate a timing (clocking) pulse. R_1 is adjusted so that $E_a = 0.0$ when the eye-lid is in the normal position.

To produce the clocking pulse a two-valued sensor was used. The output of this device was logical one when $E_a > 0.0$ and logical zero when $E_a < 0.0$. A high gain D.C. amplifier driving a relay was used as the two-valued sensor because of the small hysteresis effect as a result of high gain (15K). The pulse resulting from relay action was differentiated approximately by an R-C network, Fig. 5, with a time constant of 3 μseconds. The negative absolute value of the differentiator output (E_d) produced the series of pulses shown in Fig. 7. E_d was the input to a clamp device (its purpose is to hold the value of E_d = logical zero when no pulse is present). The output of the clamp (E_c) is shown in Fig. 6. (See Appendix B for the circuit designs and calculations.) Various waveforms throughout the system are shown in Fig. 7.

It is important to note that the clamp circuit will provide a measure of pulse shaping if the input sensitivity is set correctly. In this case, the sensitivity was adjusted (see Appendix B) so that the clamp output would be logical one for 63 percent of the input pulse. (See Fig. 6.) This will prove to be important.

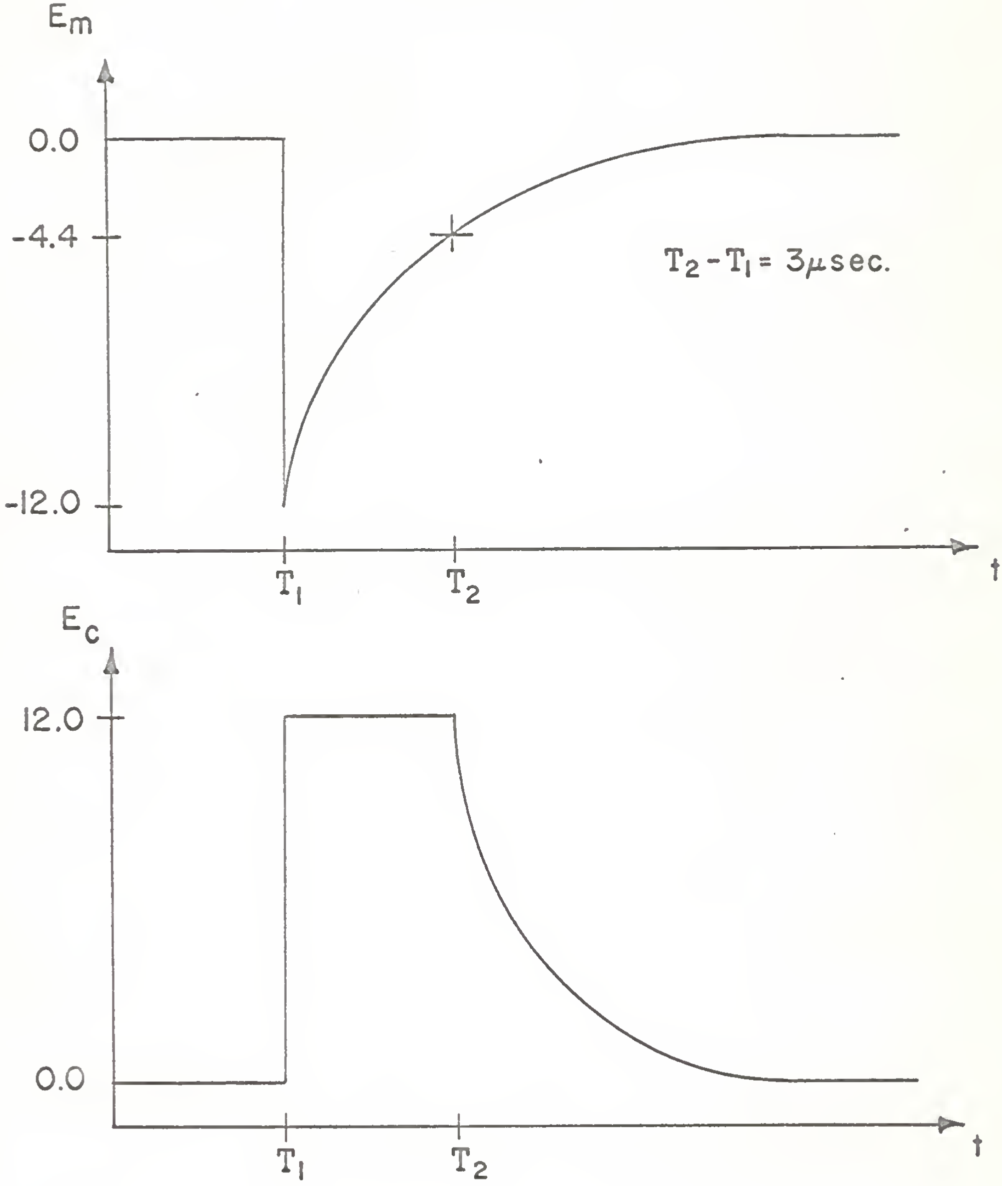


FIG. 6. CLAMP CIRCUIT PULSES

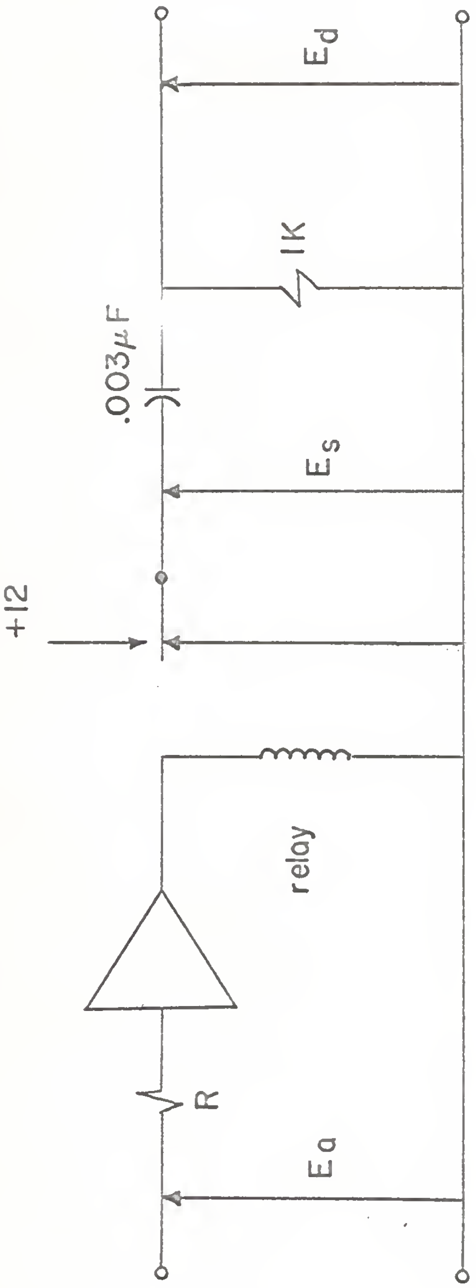


FIG. 5. TWO-VALUED CIRCUIT

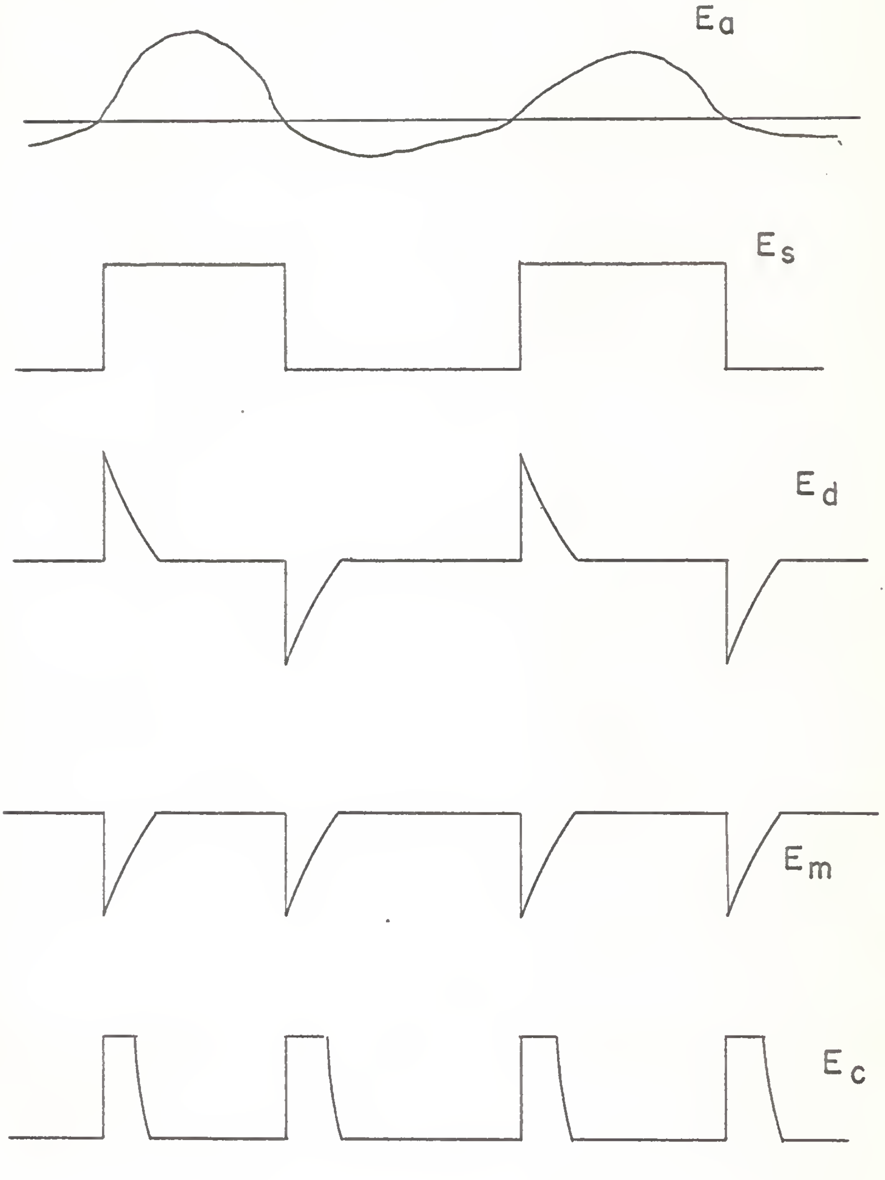


FIG. 7. WAVEFORMS

Logic Design

The problem at hand was to design a logical network that would integrate and maintain separately $\int E_e dt$ each time it crossed the baseline V_{bL} up to a maximum of three areas. From this statement it is obvious that the device must be capable of attaining seven states, changing states with the clocking pulse, as indicated:

1) Power on; system at rest.
2) Integrator 1 on.
3) Integrator 1 off.
4) Integrator 2 on.
5) Integrator 2 off.
6) Integrator 3 on.
7) Integrator 3 off; system must stay in this state until manually reset.

These states are plotted on a flow diagram with all information about each stable state noted below that state.[1] (See Fig. 8.)

A binary counter was used to count the appearances of the clocking pulses (up to seven). The counter was composed of set-reset type flip-flops. The flip-flop outputs were gated to drive relays in the inputs of three analog integrators. The number of flip-flops required is N, where N is determined by Equation 2 (assuming no redundant states are present).[2]

[1] Reference 1, p. 620.

[2] Reference 1, p. 496; Reference 2, p. 184.

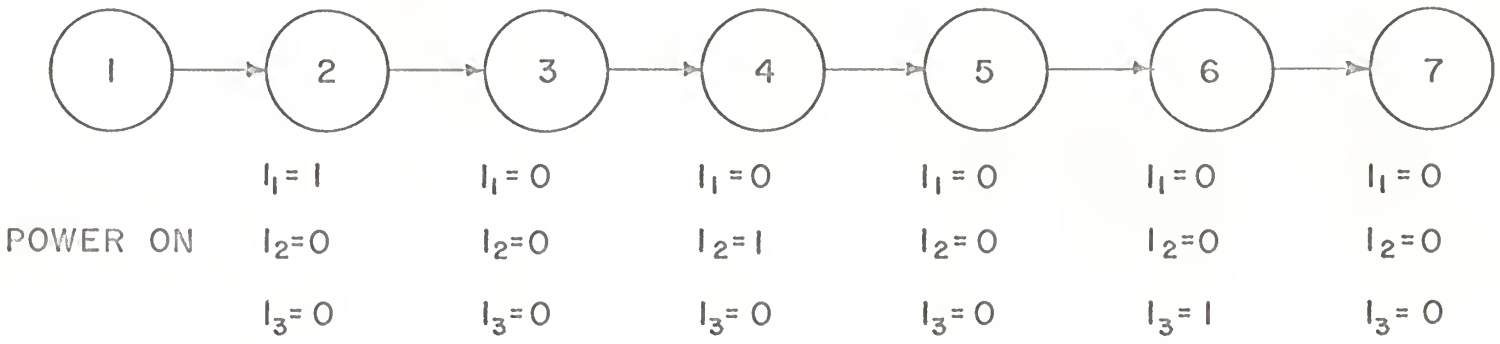


FIG. 8. FLOW DIAGRAM

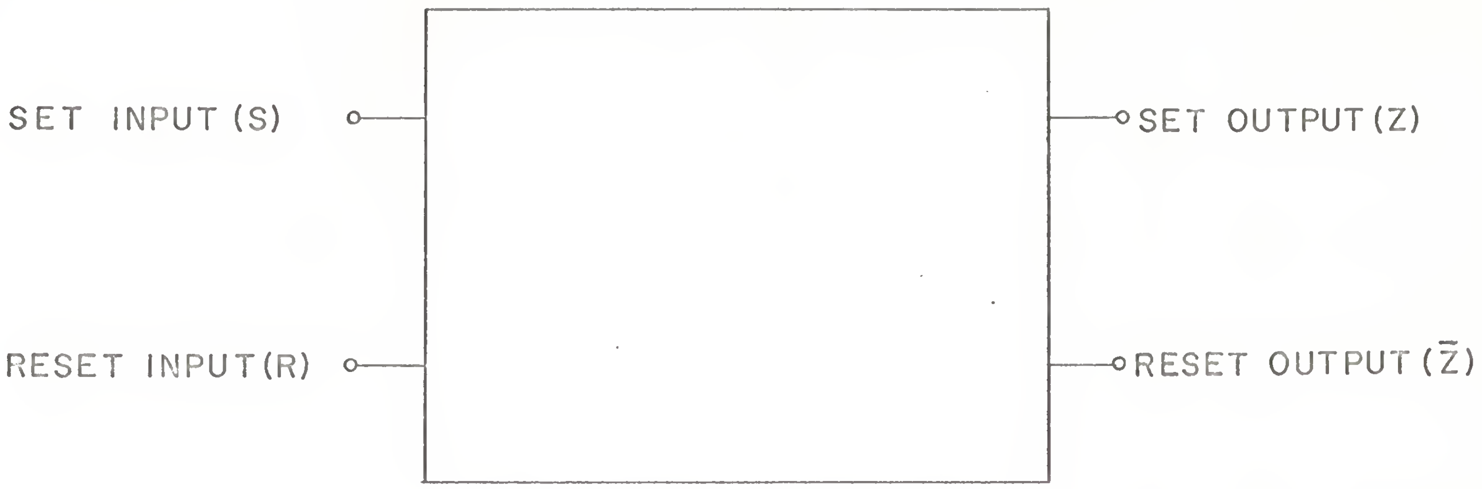


FIG. 9. SET-RESET FLIP-FLOP

$2^N \geq$ number of states 2)

The flip-flops are shown diagramatically in Fig. 9. The operation of the flip-flop can be described as follows:

1) When the set node (terminal) is pulsed, the output, Z = logical one, where one is a positive voltage level and $\overline{Z}$ = logical zero.

2) Pulsing the reset node forces Z = logical zero and $\overline{Z}$ = logical one.

3) The restriction that the set and reset nodes cannot be pulsed simultaneously is made.

The flip-flop action can also be described, by

$Z' = S + \overline{R}Z$ 3)

$SR = 0$ 4)

or by their complements,

$\overline{Z}' = S(\overline{R} + \overline{Z})$ 5)

$\overline{S} + \overline{R} = 1$ 6)

where

Z' (or $\overline{Z}'$) refers to the next state of an output.

Before the actual design work can proceed, it is important to investigate counter-action using flip-flops. The reason for doing so is anticipation of problems that will arise.

Suppose we wish to design, by trial and error, a simple counter which will count only two digits, zero and one and it is to repeat this action indefinitely. Transitions can occur only when a clocking pulse (x) occurs. If the states of a flip-flop

PRESENT STATE	NEXT STATE
Z	Z'
0	1
1	0
0	1
1	0
0	1
1	0
0	1

FIG. 10. STATE TABLE

meeting these requirements are put in a table, the following conclusions can be drawn: (See Fig. 10.)

1) Z goes to one only when Z has just previously been zero.

2) Z goes to zero only when Z has just previously been one.

From these conclusions we can determine the combinational logic expression required at the set and reset nodes. The expressions are:

$$S = \overline{Z} \text{ and } R = Z \qquad 7)$$

This results in the connection in Fig. 11. At this point one may chose to walk (by assuming clocking pulses) the circuit through several cycles. At first glance the conditions appear to be satisfied but further consideration will show that this may not be so. Assume the circuit is in state Z = 0 and the clocking pulse (x) is absent. The S gate will accept the next pulse. If the flip-flop changes states faster than the duration of the clocking pulse, Z will equal one while the clocking pulse is still present. Note that the gate at R is receptive and the R node will be pulsed. This will cause S and R to be pulsed simultaneously; this is not allowed and indeterminate action will result. To correct this difficulty, a delay is placed between the gates and the input nodes to store the pulses for the duration of the clocking pulse. This modification is shown in Fig. 11. The required delay time, according to Gray (4), is the minimum amount which will result in the proper circuit action.

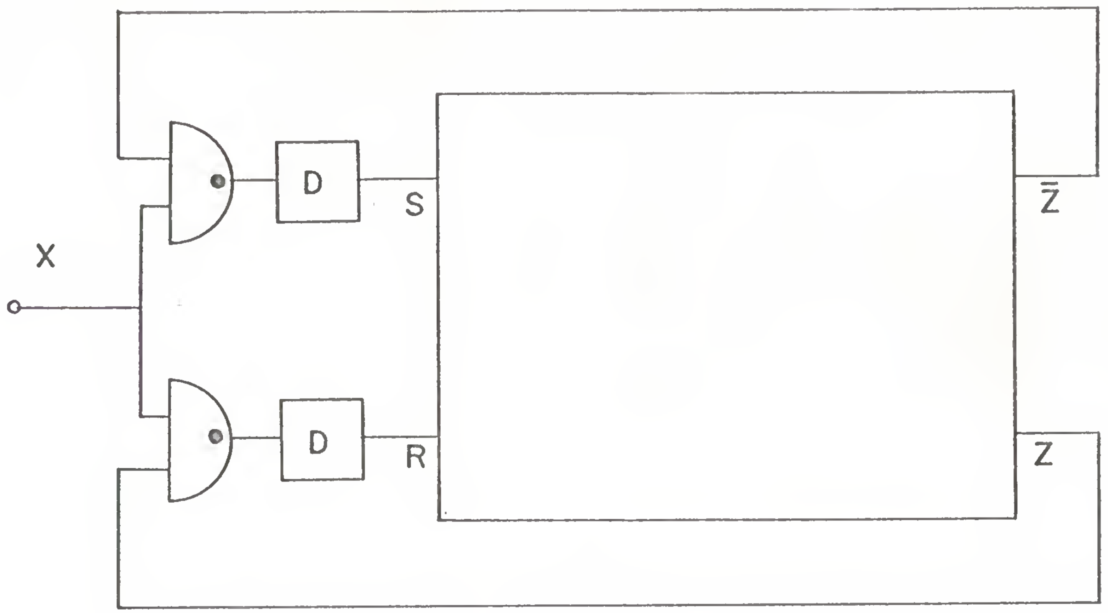


FIG. 11. TWO-BIT COUNTER

COUNTER DESIGN

To determine the logical interconnection of the flip-flops, the method presented by Phister (3) will be incorporated. Recalling that three flip-flops are required and that Z' ($\overline{Z}$') represents the next state of the output, the table in Fig. 12 can be constructed. The entries in the "Z" column are determined by arbitrarily assigning any desired combination of non-repeating states. Note that the entries in row eight are identical with those of row seven, therefore, once the counter attains state seven (110), it will remain in that state until manually reset. The entries in the next-state column are determined by entering the state which the counter assumes next. The column on the extreme right contains entries which represent the control conditions of the three integrators, i.e. integrator one is on when a one appears in column one, etc. State (111) is not used and will be referred to as an optional state.

To determine the combinational logic expression for the inputs of the flip-flops, it is necessary to examine only one node (set or reset) on one flip-flop at a time. For S_1 (expression for the set node of flip-flop 1) it is necessary to examine those rows in which Z_1 goes from zero to one. This involves only row four (011). The next state is (100). Examining these two entries, it is apparent that Z_1 can go to one only if the counter has just been in the state (011). Therefore, the input

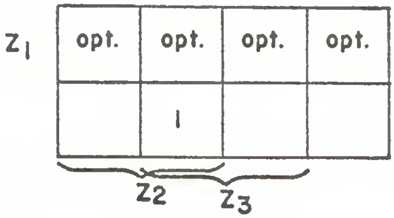
	$Z_1 Z_2 Z_3$	$Z'_1 Z'_2 Z'_3$	$I_1 \; I_2 \; I_3$
1	0 0 0	0 0 1	0 0 0
2	0 0 1	0 1 0	1 0 0
3	0 1 0	0 1 1	0 0 0
4	0 1 1	1 0 0	0 1 0
5	1 0 0	1 0 1	0 0 0
6	1 0 1	1 1 0	0 0 1
7	1 1 0	1 1 0	0 0 0
8	1 1 0	1 1 0	0 0 0

FIG. 12. STATE TABLE

expression for S_1 is $\overline{Z}_1Z_2Z_3$. It would be desirable to simplify this expression if possible. Taking into consideration rows five, six and seven, one can see that Z_1 remains one throughout these transitions. It is obvious that the set node (S_1) could be pulsed any number of times after it has previously been pulsed once. Therefore, entries in rows five, six and seven can be used as optional states to simplify the combinational expression for S_1. Recalling that state (111) does not occur (the counter can never assume this state), it can also be used to simplify S_1. The complete expression (with optional terms underscored) is:

$$S_1 = \overline{Z}_1Z_2Z_3 + \underline{Z_1\overline{Z}_2\overline{Z}_3} + \underline{Z_1\overline{Z}_2Z_3} + \underline{Z_1Z_2\overline{Z}_3} + \underline{Z_1Z_2Z_3} \qquad 8)$$

Simplification by a Karnaugh map yields (See Fig. 13.)



Z_1	opt.	opt.	opt.	opt.
		1		

FIG. 13. KARNAUGH MAP

$$S_1 = Z_2Z_3 \qquad 9)$$

To determine the expression for a reset node (R_2), examine the rows in column Z_2 in which Z_2 goes from one to zero (row four). The rows in which Z_2 is zero and remains zero are rows one and five which are optional entries. Recall that state (111) is always optional. The simplified combinational expression for R_2 is

$$R_2 = Z_2Z_3 \qquad 10)$$

A similar procedure can be used to determine the expressions for the other input nodes. They are

$$S_1 = Z_2Z_3 \qquad 9)$$

R_1 - nothing required 11)

$$S_2 = \overline{Z}_2Z_3 \qquad 12)$$

$$R_2 = Z_2Z_3 \qquad 10)$$

$$S_3 = \overline{Z}_3\overline{Z}_1 + \overline{Z}_3\overline{Z}_2 = \overline{Z}_3(\overline{Z}_1 + \overline{Z}_2) \qquad 13)$$

$$R_3 = Z_3 \qquad 14)$$

The factored form of S_3 (Eq. 13) will be used because of the reduction of logic elements. Note that the time delay of this expression does not equal that of the other expressions.

The output logic is written directly from the present state column of the state table (Fig. 12). The output expressions are

$$I_1 = \overline{Z}_1\overline{Z}_2Z_3 \qquad 15)$$

$$I_2 = \overline{Z}_1Z_2Z_3 \qquad 16)$$

$$I_3 = Z_1\overline{Z}_2Z_3 \qquad 17)$$

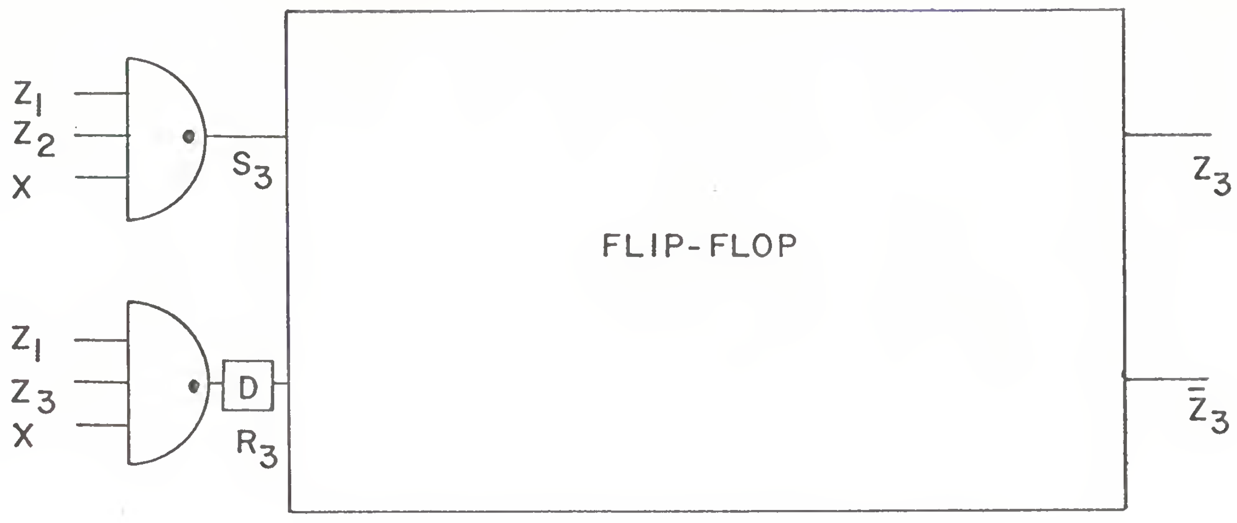


FIG. 14. ILLUSTRATION OF DELAY REQUIREMENTS

The logic expressions for the inputs and outputs were realized using diode resistance logic. The design of these elements (and the flip-flops) is found in Appendix B. The basic logic design has now been completed and the problem of the delays will now be treated.

DELAYS

As previously stated, the optimum delay time is the minimum amount that will result in proper circuit action. In more specific terms:

$$D = T_x - T_s \qquad 18)$$

where,

T_s = switching time of flip-flops

T_x = width of clocking pulse

In this case, the switching time of the flip-flops was considered to be negligible. ($D \doteq T_x$.)

Referring to the previous discussion of delays, it is remembered that the delays were placed in the input line of both the set and reset nodes. This is not always necessary.

The following statements give conditions for which a delay is required:

Assume that the switching time of the flip-flop is less than the width of the clocking pulse. Then a delay must be placed in the input line of the flip-flop if that input is dependent on the present state of the output of that flip-flop. (Refer to Fig. 14.) If the switching time of the flip-flop is greater than the width of the clocking pulse, no delay is required.

The actual delay was formed by differentiation of the pulse that is formed at the gate by the clocking pulse. Recall that this pulse, which was shaped by the clamp, has the form shown in Fig. 15 a. After negation and differentiation, the pulse will be of the form shown in Fig. 15 b. The trailing edge of the original pulse will form the positive pulse delayed 3 seconds, (which is the width of the original pulse), if the time constant of the differentiator is 1/3 or less than that of the differentiator producing the clocking pulse. The time constant of the differentiator must be sufficiently small to prevent supression of the positive pulse. The time constant chosen was 1 second. (Note, the pulse energy must be sufficient to drive the flip-flops.) From the above it is obvious that no delay is needed in the set or reset line of flip-flop 1.

Races and Hazards

According to Marcus (2), a race is defined as a situation that arises when a sequential circuit makes a transition, to one or more invalid final states. An example is:

Suppose the counter just designed was in state (101) and a clocking pulse appeared. The counter should go to (110) but under certain conditions might go to some other state. These conditions are represented in Fig. 16. Z_2 and Z_3 are two inputs,

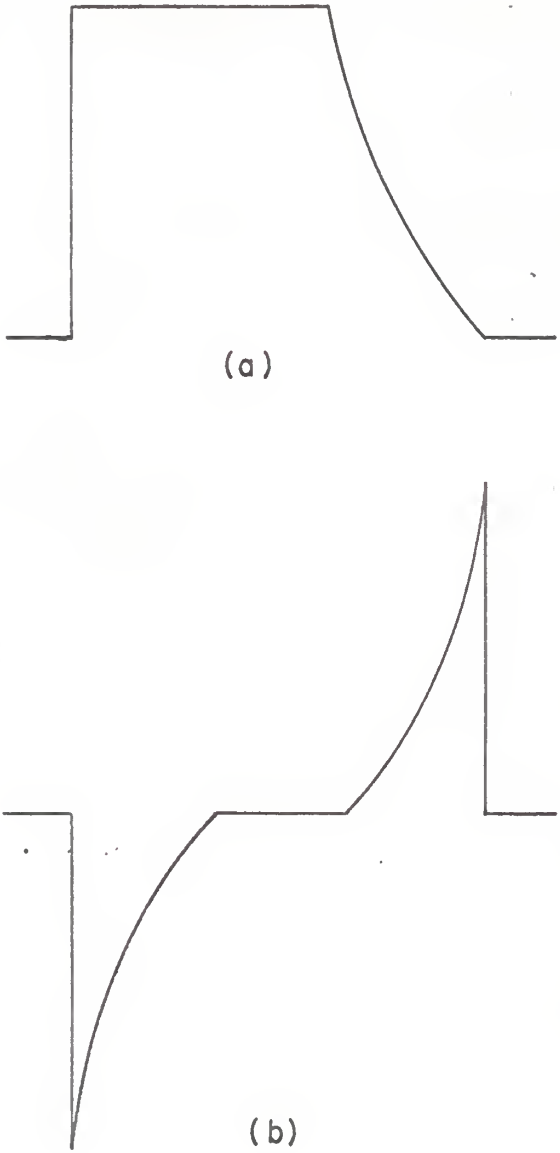


FIG. 15. DELAY PULSES

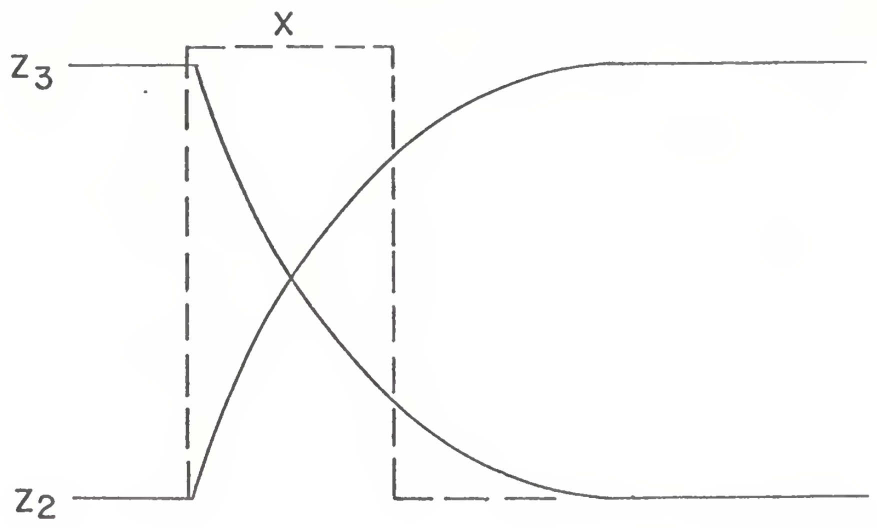


FIG. 16. RACE CONDITIONS

x, the clocking pulse. Assume that at the time of the clocking pulse Z_3 goes to zero, and Z_2 goes to one. A spike will be generated that might have sufficient amplitude to trigger the flip-flop. To eliminate the race (extra pulse that has been generated) several approaches are available:

1. Improve the transient response of the flip-flops. (The outputs of the flip-flops are A and B.)
2. Decrease the width of the clocking pulse.
3. Decrease the sensitivity of the flip-flop so that the spike will be ignored.

Probably the best solution to the race problem is to decrease the clocking pulse width and to improve the transient response of the flip-flops.

A hazard is defined by Caldwell (1) as a loss of continuity. This problem will not arise if the switching time of the flip-flops plus delay (ignoring any effects of coupling networks) is less than the interval between clocking pulses. This particular problem did not arise.

The above mentioned situations, no matter how small, can result in improper operation of the circuit if they are ignored.

Interconnection of Logic Elements

The major portion of this section will concern Eqs. 9 through 17. From these equations and a knowledge of Boolean combinational expressions, the block diagram of Fig. 17 can be constructed. The combinational expressions for the R_n's and

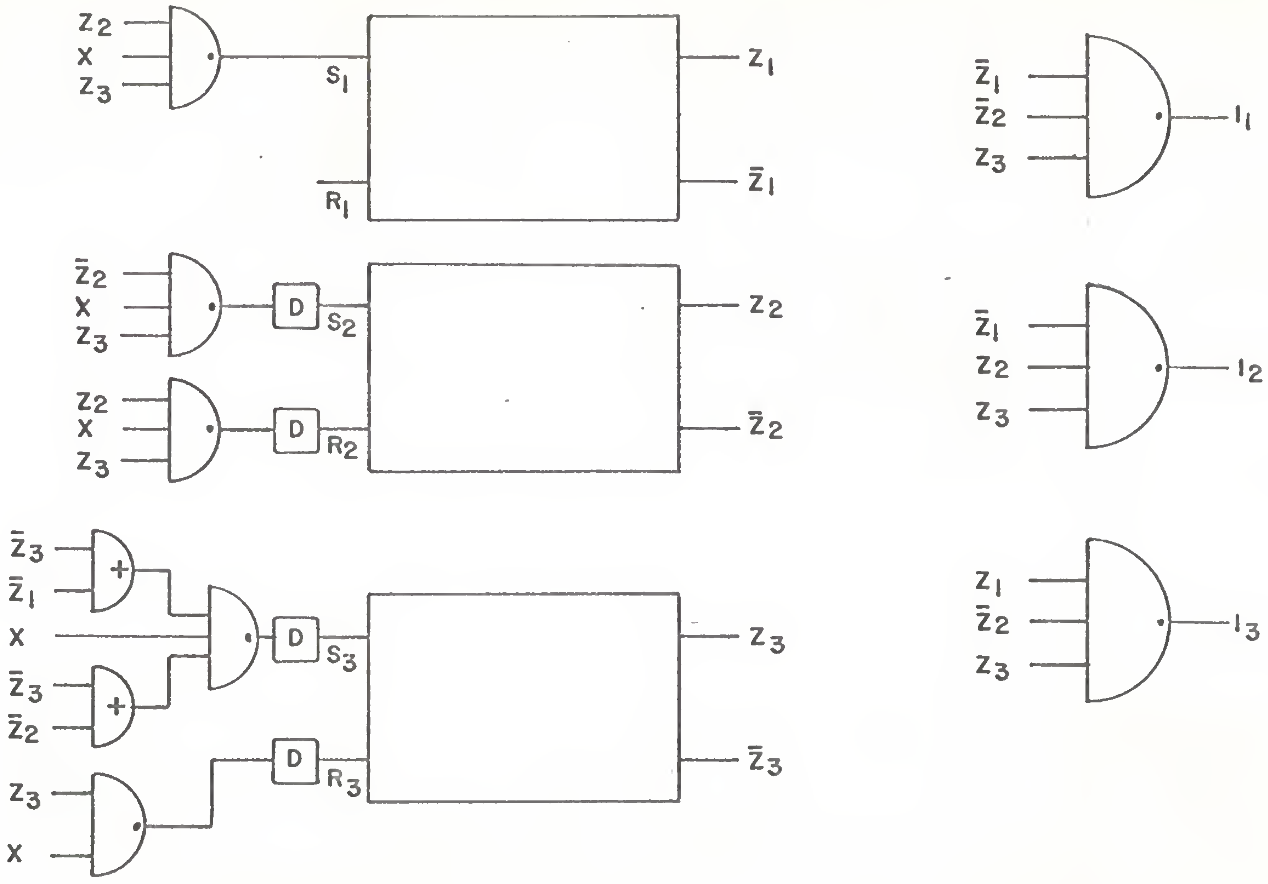


FIG. 17. SYSTEM DIAGRAM

the S_n's are placed in the input lines of the flip-flops. Note the expressions for S_1 and R_2 are equivalent and those nodes may be connected together. The clocking pulse is included as an input to each "and" gate. The outputs are formed by gates which drive relays in the input circuits of analog integrators. The signal to be integrated is E_a (signal proportional to eye-lid movement).

A provision must be made to record the time at which E_a rises above zero. This is accomplished by driving a transistor (which is in shunt with a relay) with the output gate signals. The circuit is shown in Fig. 18.

The design of specific logic elements is found in the appendix.

Modifications

Assignments of the flip-flop states in Fig. 13 were made arbitrarily. Investigation of assignment order shows that certain simplifications can be made.

The flip-flop states are altered to form the state table of Fig. 19. Note that assignments are made so that not more than one flip-flop output can change state at a time. This assignment results in the following input combination logic expressions:

$S_1 = Z_2Z_3$ 19)

R_1 - nothing required 20)

$$S_2 = Z_3 \qquad 21)$$

$$R_2 = \overline{Z}_3 \qquad 22)$$

$$S_3 = \overline{Z}_1\overline{Z}_2 \qquad 23)$$

$$R_3 = Z_1 \qquad 24)$$

A brief examination of the above expressions shows that the input to each node is independent of the output of that particular flip-flop. If the width of the clock pulse is less than the switching time of the flip-flop, the result (from previous work) is the elimination of delays. A reduction in required logic elements is also effected. The inputs to the integrators (output of logic circuitry) are determined as before. At this time, one should be aware that the above simplification cannot always be made and should be regarded as a special case (certain two-state change assignments could be required).

Another modification that is now considered is that of using trigger flip-flops instead of set-reset type flip-flops.[1] One method of design using trigger flip-flops would be to use only the required input logic expressions. (The one entries on the Karnaugh map.) The circuit timing would be simplified, but input pulse levels become critical.[2] Internal logic would

[1] Reference (5), p. 305.

[2] Ibid.

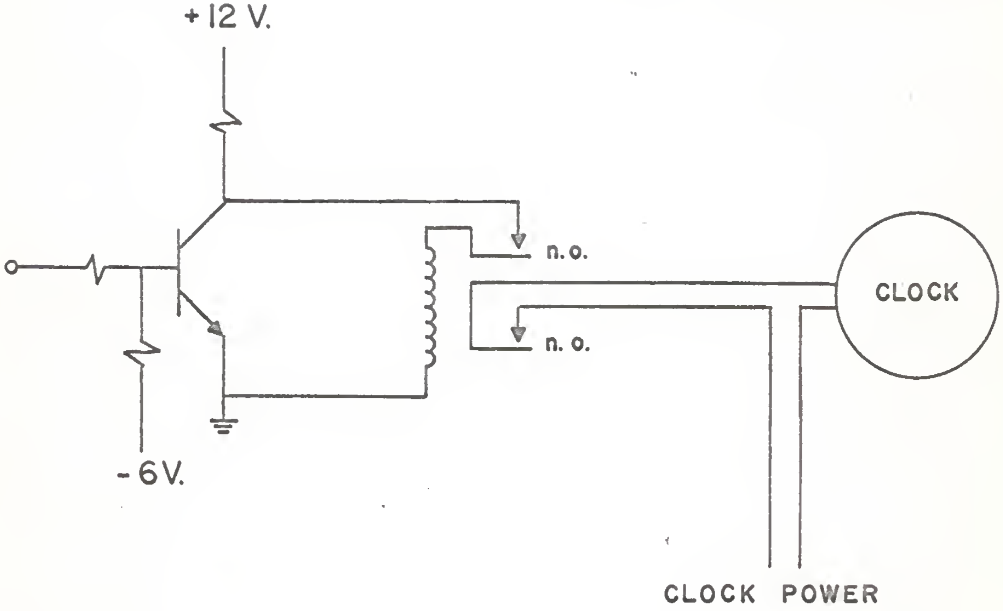


FIG. 18. TIMING CIRCUIT

$Z_1 Z_2 Z_3$	$Z_1' Z_2' Z_3'$
0 1 0	0 0 0
0 0 0	0 0 1
0 0 1	0 1 1
0 1 1	1 1 1
1 1 1	1 1 0
1 1 0	1 0 0
1 0 0	1 0 0
1 0 0	1 0 0

FIG. 19. MODIFIED STATE TABLE

be added to the flip-flops and more logic elements would be required (under certain assignments). Other modifications such as inhibiting the counting cycle by an external level can be realized by gating the input pulse with the external level.

SUMMARY AND CONCLUSIONS

In the study of sequential logic design work, great use is made of the assumed or canonical form. In this paper, it has been assumed that the basic form of the sequential logic machine is that of a simple counter. For the system discussed, the counter has only one mode of operation; that is, it counts only a predetermined fixed sequence according to the state assignments. In more sophisticated systems, the counting sequence can be altered at any time by a change of an external input or past states.

The method of design presented in the text has one outstanding feature: The design task is simplified by determining the combinational logic expression for only one particular input (set or reset node) at a time. The advantage over intuitive design procedures is obvious.

The width of the input pulse (clocking pulse) is an important factor. As previously discussed, it contributes considerably to the timing difficulties. The best circuit action can be obtained by making the clocking pulse narrower than the switching time of the flip-flops. (Of course, it must be of sufficient duration to trigger the flip-flops.)

Referring to the Appendix B, it will be noted that saturated design was used extensively. In this particular application, measurement intervals less than 15 milliseconds are considered to be negligible.

In the section pertaining to modifications, it was remarked that if the counter states could be assigned to allow only one change of variable (flip-flop output) per transition, the required number of logic elements would be reduced. This method of state assignment is called by Phister (3) and others as the Gray Code assignment.[1] It is important to point out that the Gray Code has many advantages in counter designs but has disadvantages in computing. (A Gray Code number must be converted to binary if an arithmetic operation is to be made on that number, unless the entire logic system is in terms of a Gray Code.)

Previously mentioned in the introduction of the paper was the fact that sequential logic design procedures cannot be expressed in general but must refer to a particular physical system. Although this may not be true of trivial systems, the size and complexity of others prohibit the derivation of

[1] Reference (3), p. 232.

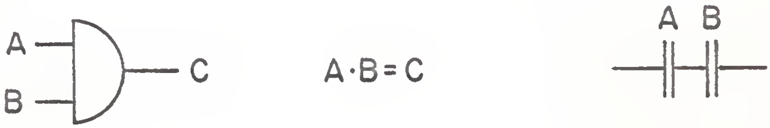
general design equations. The alternative approach to the design of a complex and large physical system is to study and apply these procedures in a straight-forward and logical manner.

The system described in the text is now being used to collect and score data by the Psychology Department at Kansas State University. No difficulties have arisen to this date.

APPENDIX A

Symbols and Definitions

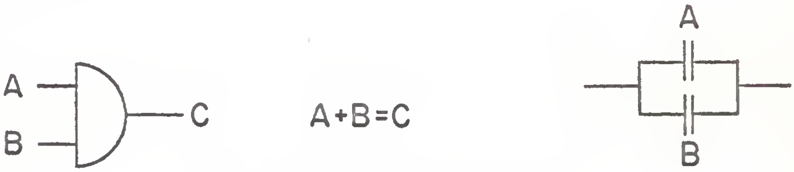
Positive two-valued Boolean logic, as used by Caldwell (4), is used throughout the text. The "and" operation (series connection) is represented algebraically by multiplication, and symbolically by



$$A \cdot B = C$$

Fig. 20. AND OPERATION

The "or" operation is represented algebraically by addition and symbolically by



$$A + B = C$$

Fig. 21. OR OPERATION

$\bar{Z}$ is defined as not Z. Z' is defined as the next state of Z. S and R are the set and reset nodes of a set-reset flip-flop.

APPENDIX B

Circuit Design

The logic elements and pulse forming networks are designed using transistors. The transistors used were type NPN 1304. All designs make use of the equivalent circuit shown below. Only the DC beta, base to emitter voltage, and the maximum ratings are assumed. The operational amplifiers are G. P. Philbrick, Researchers, type K-2W.

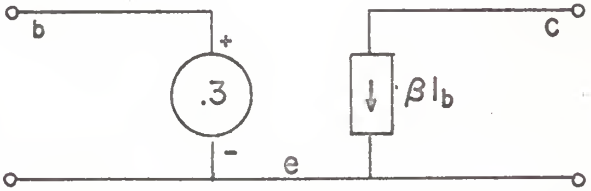


Fig. 22. TRANSISTOR EQUIVALENT CIRCUIT

The circuits and the design equations will be shown in the order of presentation.

(1) R-C Differentiator.

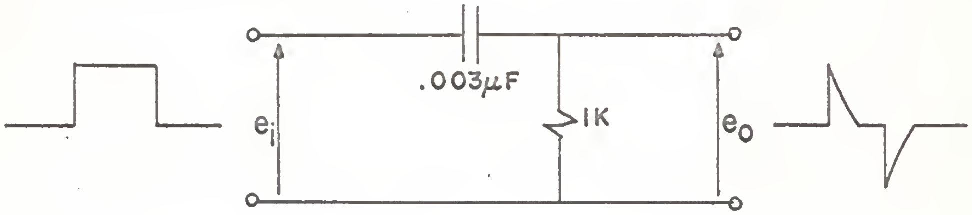


Fig. 23. R-C DIFFERENTIATOR

The load, Z_L, is the input to another transistor and is equal to 10K. The coupling capacitor, C, is chosen so that the time constant of the coupling network is somewhat larger than the width of the pulse, but shorter than the interval between input pulses (which is never less than 100μseconds).

(3) Clamp Circuit

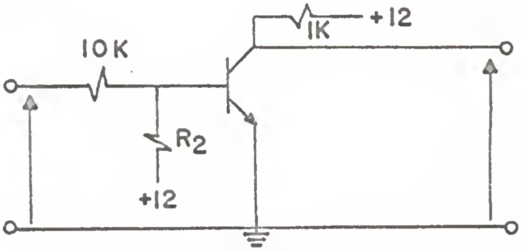


Fig. 25. CLAMP CIRCUIT

The operation of the clamp circuit is desired to produce a measure of pulse shaping. The input pulse has increased to -4.5 volts in 3μ seconds. The value of R_2 is determined so that the transistor will be in saturation at E_i = -4.5 volts.

$$-(4.5 + 0.3)/10 + 11.7/R_2 = 0.24 \text{ m.a.} = I_b \qquad 31)$$

$$R_2 = 16k \qquad 32)$$

The input and output waveforms are shown in the diagram.

(4) Flip-Flop Circuit.

The two flip-flop transistors, Q_1 and Q_2 are cross coupled "not" circuits. (See Fig. 26.) The design equations are determined by considering only one transistor at a time, and applying nodal analysis, as before, we get:

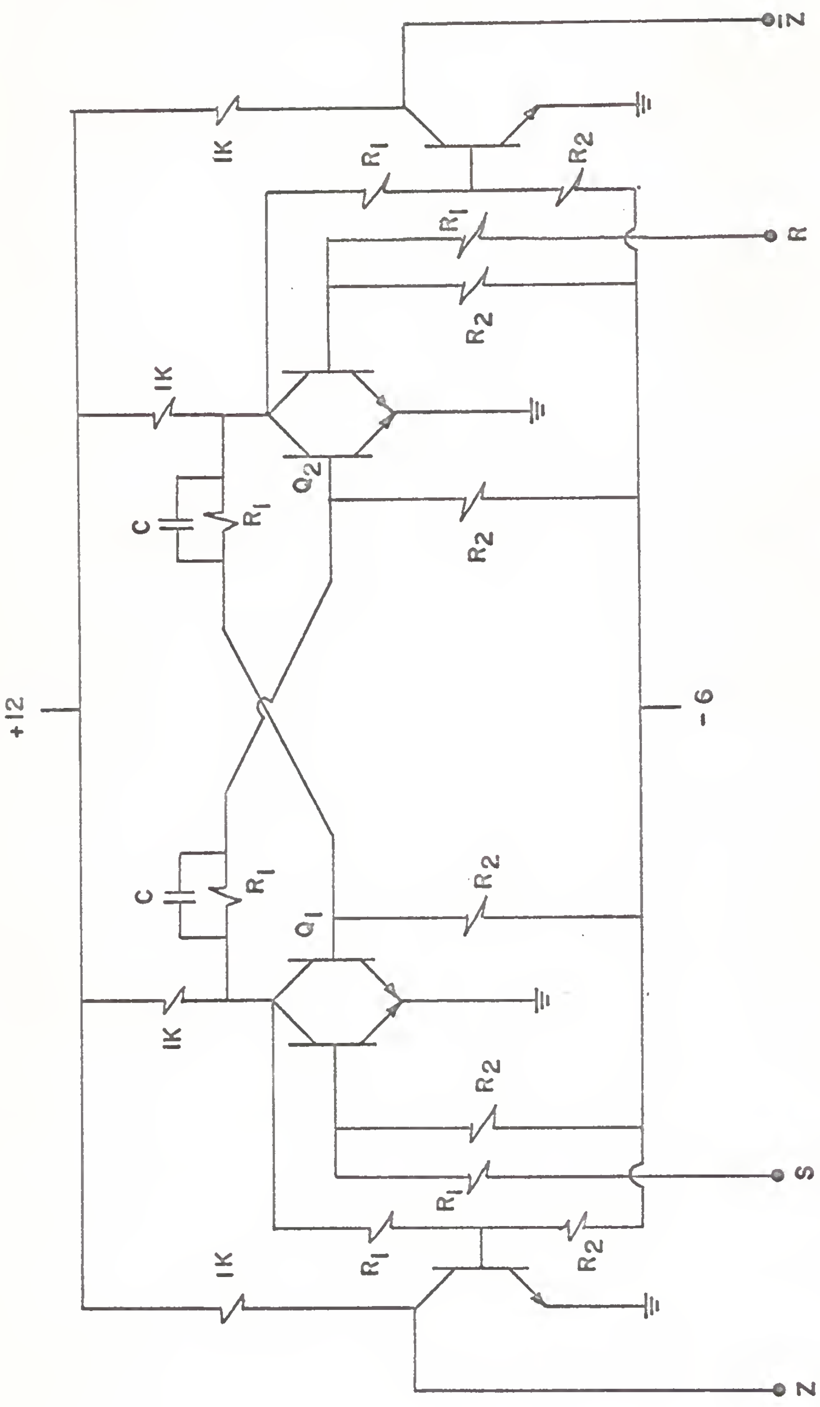


FIG. 26. FLIP-FLOP CIRCUIT

$R_1 = 10k$ 33)

$R_2 = 4k$ 34)

$c = .0001 \quad F$ 35)

The "speed up" capacitors (c) are used to improve the transient response. An improvement will obviously be made due to the fact that the network has an initial output voltage when the flip-flop changes state.

(5) "And" Gate.

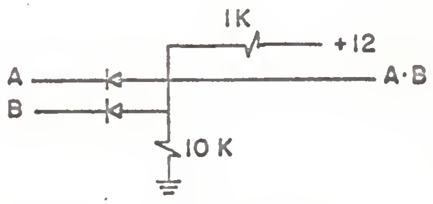


Fig. 27. "AND" GATE

The values of the resistances are chosen so that 90 percent of the input pulse will appear at C when A = B = 1.

(6) "Or" Gate.

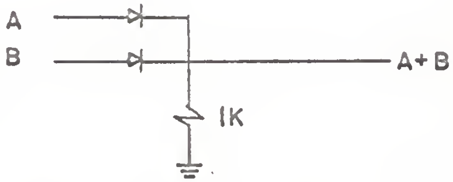


Fig. 28. "OR" GATE

(7) Delay Circuit.

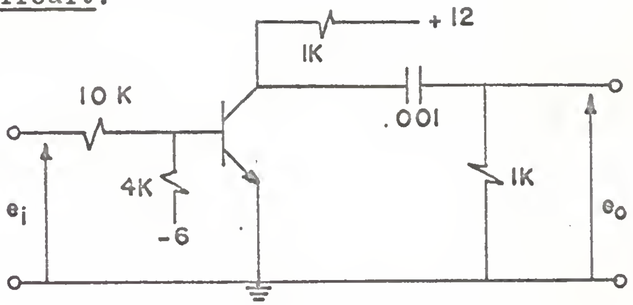


Fig. 29. DELAY CIRCUIT

The purpose of the delay is to store a pulse for a duration equal to the width of the clocking pulse. The transistor inverts the input pulse which is approximately differentiated by the R-C network. The time constant of the R-C differentiator is 1 second. This allows the capacitor to discharge before the trailing edge of the input pulse appears. The desired output is the positive pulse, generated by the approximate differentiation of the trailing edge of the input pulse since the logic circuit does not recognize negative voltages. The design equations are identical to Eqs. (28) and (29).

The Laplace transform of $e_i(t)$ is

$$E_i(s) = (1-e^{-TS})/S \qquad 25)$$

where $T \geq 30\ \mu$seconds, and the Laplace transform of the output voltage is

$$E_o(s) = E_i(s) \cdot \frac{R}{R + 1/CS} \qquad 26)$$

Since the pulse width is more than 30μ seconds, the output, E_o, will be as shown in the figure. (The capacitor has time to discharge before the trailing edge of the pulse appears.)

(2) Negative Absolute Value Circuit.

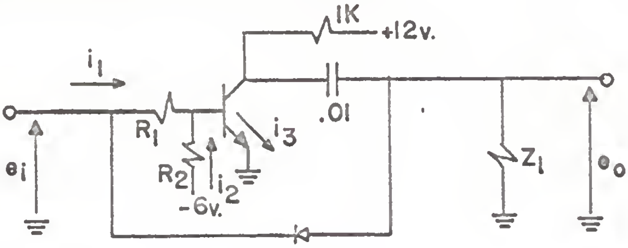


Fig. 24. NEGATIVE ABSOLUTE VALUE CIRCUIT

Under the assumption that the base to emitter voltage = 0.3 volts and the magnitude of the input pulse is 12.0 volts:

$$I_1 + I_2 = I_3 \qquad 27)$$

The value of I_3 for saturation is

$$I_3 = 12.0/(R_L \cdot \text{Beta} \cdot 0.5) \qquad 28)$$

$$I_1 + I_2 = (12.0 - 0.3)/R_1 - (+6 + 0.3)/R_2 \qquad 29)$$

Letting R_1 = 10K to minimize loading of the differentiator and solving for R_2 one obtains:

$$R_2 = 4K \qquad 30)$$

BIBLIOGRAPHY

(1) Caldwell, Samuel, Switching Circuits and Logical Design. John Wiley and Sons, N. Y. 1962

(2) Marcus, Mitchell, Switching Circuits for Engineers. Prentice-Hall, Englewood Cliffs, N. J. 1962

(3) Phister, Montgomery, Logical Design of Digital Computers. McGraw-Hill, N. Y. 1959

(4) Gray, Harry, Digital Computer Engineering. Prentice-Hall, Englewood Cliffs, N. J. 1963

ACKNOWLEDGEMENT

The Author wishes to thank his wife for typing the manuscript, and the others who were of great help.

SEQUENTIAL LOGIC INSTRUMENTATION

by

WILLIAM T. HULL

B. S. (E. E.)
Kansas State University, 1963

AN ABSTRACT OF
A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
MANHATTAN, KANSAS

1964

A technique for the synthesis of sequential logic circuits is developed by means of the instrumentation of a classical conditioning experiment in psychology. Binary counters together with the necessary delay circuits are utilized to count a pulse sequence in order to control a signal supplied to integrating and timing circuits. Detailed analysis and synthesis of the logic circuits as actually constructed is presented.